Memo

To:	CCB
From:	Yu-Long Kung and István Bondár
Date:	June 21, 2002
Subject:	Hypocenter Location Server Release 1.0
Sponsor:	Bob Woodward
CC:	David Salzberg

Abstract

The Center for Monitoring Research (CMR) Research and Development Support System (RDSS) Hypocenter Location Server (HLS) provides a remotely accessible interface to the CMR's hypocenter location and magnitude software (*EvLoc*). HLS makes available CMR hypocenter location capabilities to the researchers in the monitoring community. HLS supports the full range of *EvLoc*'s capabilities.

HLS will not only replace *LocSAT* but add new functionalities, such as magnitude calculation and location using hydroacoustic and infrasonic phases. HLS also uses the platform-independent XML format as the data exchange format for *EvLoc* input and output database tables. Five new database tables, **elpmonitor, elsgrouptype, elssession, elsusage, and elsuseradm** are also introduced to monitor HLS activities and provide high-level usage summaries. The current HLS Release 1.0 (HLS R1.0) operates using the CMR message system; that is, via e-mails. Future enhancements will provide an interactive web interface. The baseline for HLS R1.0 are the releases RDSS_1.0.7 and RDSS_1.0.3.

In order to utilize the Message Subsystem (MSS) as the front-end interface for users, a new MSG_TYPE PROCESS is introduced to the Message Subsystem. This modification to the Message Subsystem is backwards compatible.

Statement of Objective

The objective of this proposal is to implement a remotely accessible interface to the CMR event location and magnitude computation capabilities to support DTRA contractors. Figure 1 demonstrates the HLS data flow model. We propose to install the Hypocenter Location Server, HLS R1.0, patches RDSS_1.0.7 and RDSS_1.0.3, as part of the CMR RDSS baseline.

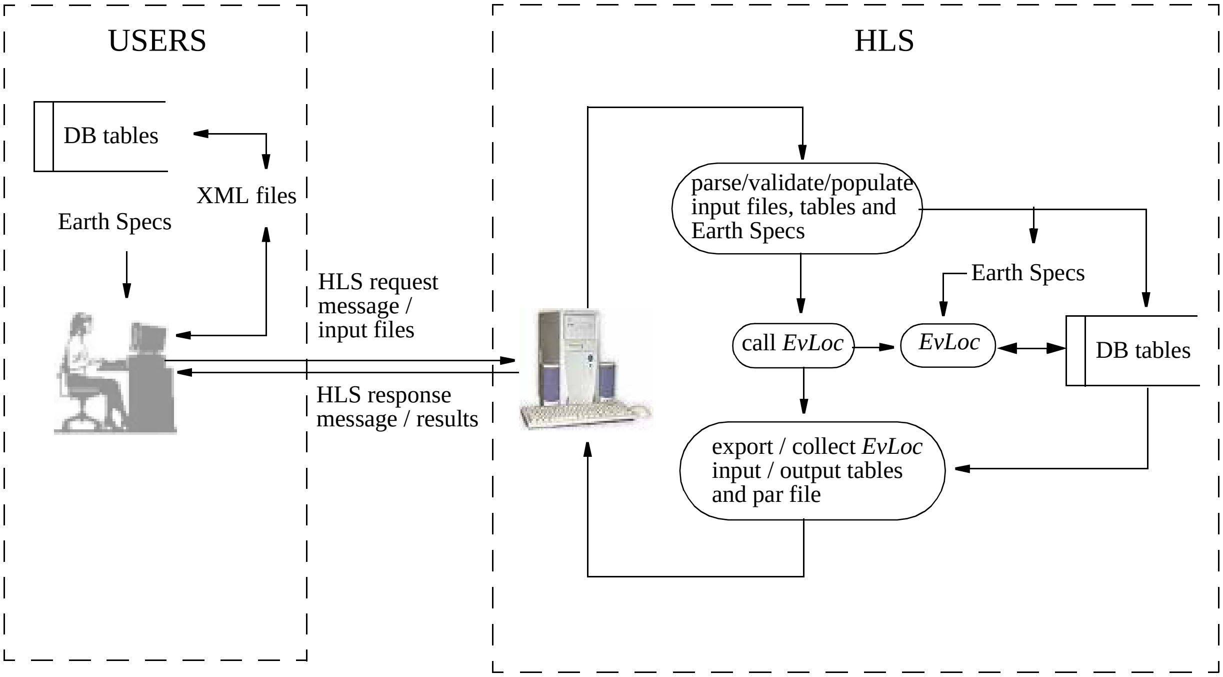

USERS
HLS
DB tables
XML files
Earth Specs
HLS request message / input files
HLS response message / results
parse/validate/populate input files, tables and Earth Specs
Earth Specs
call EvLoc
EvLoc
DB tables
export / collect EvLoc input / output tables and par file

Figure 1: HLS data flow model

Table 1 shows that the HLS server-side requirements (Appendix A) are satisfied by HLS R1.0.

Table 1: HLS server-side requirements

#	Requirement	S[1]	Comment
S1	Applies existing CMR user authentication tools for user access control	Y	
S2	Sets an upper limit on simultaneously running *EvLoc* instances and on maximum number of events that can be located by an *EvLoc* instance to avoid system overload	Y	
S3	Validates and parses input files, populates database tables and builds earth_specs directory tree	Y	
S4	Sends the requestor messages associated with HLS results, including unauthorized users, invalid input files, success, and out of service messages	Y	

Table 1: HLS server-side requirements

#	Requirement	S[1]	Comment
S5	Provides both batch and interactive event location modes	P	Interactive mode will be implemented in HLS R2.0 (web interface)
S6	Applies CMR message tracking system to allow users to refer to previously submitted data	P	Internal message tracking is implemented on the server side. The capability for incoming request messages to refer to previously submitted data has not yet been implemented. This feature will be implemented in HLS R2.0 (web interface)
S7	Generates final results in either IMS1.0 Bulletin or XML format as specified by the user	Y	
S8	Provides a mechanism to monitor *EvLoc* request/process/response status and handle exceptions	Y	
S9	Provides help information, such as user manuals, man pages, and information on data exchange formats	Y	
S10	Records usage statistics which can be used to generate reports and high level summaries on HLS	Y	
S11	Provides downloadable conversion utilities to generate required input files in XML format	Y	

[1] S means “Satisfied”: Y = Yes, P = Partial

Summary of Proposed Change

HLS is a new subsystem for the CMR RDSS. There are three aspects to the HLS subsystem: 1) new components to respond to requests for HLS services, including instantiation of *EvLoc*, 2) modifications to the CMR message handling system and 3) new database tables.

The major features and significant changes of HLS software components maintained by RDSS are described in Table 2. The new components include the HLS setup and cleanup scripts, the *EvLoc* wrapper to parse and exchange *EvLoc* input and output data, utilities to convert between

Oracle database tables and XML format flat files, and utilities to convert SSSC and travel time files from external free-format to RDSS internal format (Appendix E).

Table 2: HLS Components

Software Component	Code Type	New	Client Side Utility	Comment	Latest Release
MessageReceive	C	No		Added a new message type “PROCESS”	RDSS_1.0.3
EvLocProcess	C	Yes		*EvLoc* wrapper to parse and populate *EvLoc* input files to the proper locations and export *EvLoc* output tables in flat files in both XML and IMS 1.0 Bulletin formats	RDSS_1.0.7
DBtoXML	C	Yes	Yes	UNIX:Oracle Database table to XML-format flat file utility	RDSS_1.0.7
XMLtoDB	C	Yes	Yes	XML-format flat file to UNIX:Oracle Database table utility	RDSS_1.0.7
sssc2rdss.pl	Perl	Yes		SSSC external free-format to RDSS internal format conversion utility	RDSS_1.0.7
tt2rdss.pl	Perl	Yes	Yes	Travel time external free-format to RDSS internal format conversion utility	RDSS_1.0.7
HLS_setup.pl	Perl	Yes		HLS tables and configuration setup script	RDSS_1.0.7
HLS_cleanup.pl	Perl	Yes		HLS tables and configuration clean up script	RDSS_1.0.7
libdbxml	C	Yes		Interfaces between UNIX:Oracle database table and XML format flat file	RDSS_1.0.7

The message system has been modified to support a new message type ‘PROCESS’. Specifically, *MessageReceive* required modification to recognize ‘MSG_TYPE PROCESS’. In addition, the *MessageGet* parameter file is configured to support the new message type.

Five new database tables, **elpmonitor, elsgrouptype, elssession, elsusage, and elsuseradm** are introduced to monitor HLS activities and usages. The table descriptions, table-relationships, and table field attributes are listed in Appendix B. All of the tables have been reviewed for database schema consistency and compliance.

Additional details and manual pages are provided in both releases, RDSS_1.0.3 and RDSS_1.0.7.

COTS Software

In order to support various archived and/or compressed types of HLS input FTP_FILE such as .zip, .tar, and .tar.gz, we propose to use the archive utility *StuffIt* (version 5.2 or later) released by Aladdin Systems. The utility *StuffIt* includes two commands “stuff” and “unstuff” to create and expand an archived and/or compressed file, respectively.

Expected Benefits

HLS provides a remotely accessible interface to the CMR event location and magnitude computation facilities for the research community, without distributing any source or executable codes (with the exception of some simple format translation utilities). We provide a user guide (Appendix E) and an examples guide (Appendix F). The release of the HLS will also allow us to eliminate the maintenance effort and support for *LocSAT.*

The additional benefits are as follows:

- provide backwards compatibility with *LocSAT* (*LocSAT* input file format is supported)
- provide fully configurable *EvLoc* environment to users without requiring special knowledge of *EvLoc* directory trees and database structures
- allow magnitude determination
- support hydroacoustic and infrasonic event locations
- implement platform-independent data exchange via XML-format files
- eliminate the need for distributing source codes and executables

Possible Risks and Dependencies

The RDSS_1.0.3 Message Subsystem (MSS) introduces the new message type “PROCESS”. If this version of the MSS receives a message of type PROCESS, but an HLS is not installed, the MSS will send an alert message to the operator, but the message sender will not receive any alert message from the system. Thus, this version of the MSS should only be used at the CMR, where the HLS will be installed and operating, and should not be used at other sites.

Unit testing has verified the proper operation of the MSS with respect to handling messages of all types, including the newly defined type PROCESS. However, we have not performed a full integration test of the MSS due to lack of an integration test environment. We recommend this risk be mitigated by initially running a separate instance of the MSS for just the HLS (this is discussed further in the section ‘Schedule and Plan for Implementation’). After a successful validation period (we recommend three months) we recommend running a single instance of the new MSS (RDSS_1.0.3) to process all message types.

The HLS performs various database transactions, including “drop” and “truncate”, and this obviously carries some risk. The HLS interacts with three types of database tables: 1) *EvLoc* dependent tables, 2) HLS administration tables, and 3) request message tracking tables. Only the request message tracking tables are also used by the MSS. The “truncate” transactions only take place on the *EvLoc* dependent tables when *EvLocProcess* is triggered by an HLS request message. These tables are only used by the HLS-instantiated *EvLoc*. The “drop” transaction is performed by the cleanup script *HLS_cleanup.pl*, which is manually executed by the operator. The cleanup script uses configurable parameters to drop tables which belong to each type. To minimize the risk of dropping those tables by accident when the operator executes *HLS_cleanup.pl,* it is strongly recommended to use separate database account for HLS. That is, the database tables of type 1 and 2 (as described above) should belong to an HLS database account, and all HLS processes (including operator actions) should execute as an HLS user. This will greatly reduce the risk of unintended database modifications, as the HLS will only be able to modify those tables on which it has write permission.

A final possible risk in implementing HLS R1.0 is its vulnerability to user-provided input archive files. A very large input file might quickly fill up HLS storage resources and slow down the service availability and performance. The current *MessageFTP* has the same risk as it retrieves remote files with unknown file size through the ftp protocol. We will investigate the possibility of imposing a limit on the imported archive file size before retrieval. If such a limit is feasible, we will release it as a patch. HLS R1.0 also inherits risks embedded in the Message Subsystem and *Evloc,* such as possible denial-of-service attacks on the Message Subsystem and possible *EvLoc* segmentation faults.

Summary of Testing

The binaries and libraries of the release RDSS_1.0.7 were compiled on a Sun Ultra-1 host using Solaris 7 and Oracle 8.1.5.0.0. The v5.0 C compiler was used for this build. This release was tested on Sun SPARCstation 20 using Solaris 7 and Oracle 8.1.6.0.0

During the beta test period, we collected 164 test messages. The test messages cover seismic, hydroacoustic and infrasonic locations, and magnitude determination. The test reports are listed in Appendix C. In table 3 we summarize the HLS usage statistics from the **elsusage** table among those messages. Table 4 shows the optional input file usage. The optional input files are the *EvLoc* user-defined par file (Par file), the velocity model specification file (VMSF), the magnitude description file (MDF), source specific station corrections (SSSC) files, slowness-azimuth station corrections (SASC) files and travel-time tables (TTT) files.

Table 3: Test statistics retrieved from the elsusage table

Field	Minimum	Average	Maximum	Comment
TOTALPROTIME	10.680	100.829	2131.520	Total process time in seconds, including *EvLoc* process time and all setup/cleanup house-keeping process
EVLOCCPUTIME	0.7	10.908	90.500	*EvLoc* process time in seconds
TOTALEVENT	1	7.305	32	Total number of events to be relocated
EVENTCOUNT	1	5.082	24	Number of successfully relocated events
TOTALARRIVAL	3	111.881	1712	Total number of arrivals
IFSIZE	1	2709.839	29677	Input file size in kbytes

Table 4: Number of messages which provide optional input files

	Par file	VMSF	MDF	SSSC	SASC	TTT
Number of messages	119	20	19	14	6	20

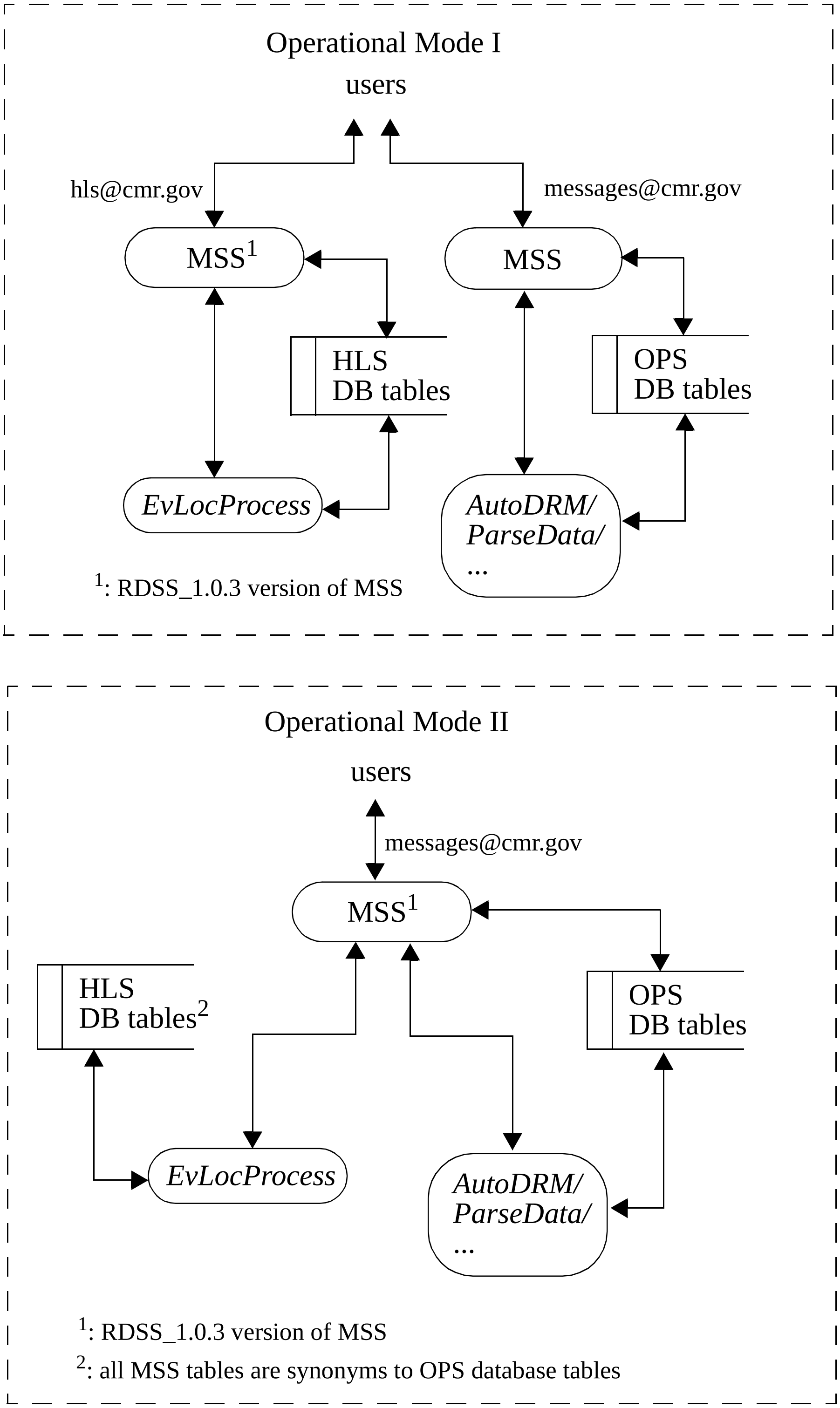
Schedule and Plan for Implementation

There are two possible operational modes for running the HLS in the CMR Operations environment, as shown in Figure 2. The mode I is to run a separate MSS instance to process HLS request messages. The mode II simply runs one MSS to process all incoming IMS messages including HLS request messages (i.e. message type PROCESS). Though the MSS released in RDSS_1.0.3 is fully backwards compatible, we propose to use the Mode I operation for a fixed period of time (three months is recommended) to observe the reliability of HLS behavior and HLS usage, before switching to the Mode II. Upon switching to Mode II it will be necessary to monitor the MSS carefully during the initial period of operation, and roll back to a Mode I style installation if problems are encountered. Regardless of the operational mode selected, we strongly recommend the use of a separate database account for HLS. All MSS tables in the HLS database are synonyms to the CMR Operations MSS database tables.

To install HLS R1.0, CMR Operations should install RDSS_1.0.3 first before installation of the patch RDSS_1.0.7. We recommend to install HLS on a Sun SPARCstation 20 or better machine. We suggest the following steps for installing and implementing HLS on RDSS.

1. Assign the environment variable ELS_ROOT to the directory pathname where HLS R1.0 will be installed and make sure that the following environment variables are defined: IMSPAR, ORACLE_HOME, ORACLE_SID, TNS_ADMIN, ORABIN, GDI_HOME, CMS_CONFIG, CMS_HOME, CMS_SHARED.
2. Decide on the number of *EvLocProcess* instances to set up based on hardware resources.
3. Revise/install all parameter files used by HLS and the Message Subsystem, listed in Appendix D.
4. Execute the setup script *HLS_setup.pl* to create HLS tables, populate administration tables, and build up Earth specification configuration files.
5. Install the new binaries, and put RDSS_1.0.3 MSS under a different folder other than /cmss/rel/bin
6. Request that infrastructure set up a trust access and security share service through OpenSSH. This service will operate between the HLS machine and ftp public server machine in order to execute security copy command “scp” for copying *EvLoc* output files outside firewall.
7. Install or make sure the archive and un-archive utilities are in place, such as “*tar*” or “*stuffit*” command.
8. Start the Message Subsystem as a separate instance from standard CMR Operations’
9. Implement Mode I on an isolated account, with changeroot
10. Investigate security options before move to Mode II
11. Announce availability to the Program Research and Development Announcement (PRDA) community

Additional installation notes are provided in the RDSS_1.0.3 and RDSS_1.0.7 release notes.



[1]: RDSS_1.0.3 version of MSS

[1]: RDSS_1.0.3 version of MSS

[2]: all MSS tables are synonyms to OPS database tables

Figure 2: Two possible operational modes for HLS

Costs and Resources Required for Implementation

We estimate that the hardware resources required under normal usage will be as follows:

Table 5: HLS hardware resources

Category	Item	Space in mbytes
Database	Administration	0.64[1]
	EvLoc tables per *EvLocProcess* instance	2.56[1]
File System	Incoming/outgoing files per request	3.94[2]
	Earth specification files per *EvLocProcess* instance	16.64
	Web resources	15.00[3]

[1] based on initial extent size of 128 kbytes and next extent size of 128 kbytes

[2] based on 164 test messages

[3] including HLS web documents and HLS public examples and utilities

The cost for implementing will be about 8 man hours with cooperation from infrastructure to set up the security file copy between the HLS machine and public ftp server machine.

Appendix A: HLS Requirements

EvLoc services Phase-1 Requirements

István Bondár and Yu-Long Kung
08/08/2001

EvLoc services can be considered as a remotely accessible RDSS location server. It supports both batch processing (through e-mail) and interactive data exchange (through web access). Besides providing all of *LocSAT*'s functionalities, it also supports interactive event location. Further functionalities, such as magnitude calculation, hydroacoustic and infrasound specific travel times, hydro blockage information, etc., will be added in subsequent development cycles.
EvLoc services does not require any changes in the operational software but will rely on existing facilities, such as Message System and *EvLoc*. *EvLoc* services will use a single database account (preferably on the RDSS database server). Each *EvLoc* instance will have its own set of tables in the database and its own config/earth_specs directory tree.

Client Side

- Provides valid input files to *EvLoc* services
- Enables web browser to support JavaScript if web interface is selected

Server Side

- Applies existing CMR user authentication tools for user access control
- Sets an upper limit on simultaneously running *EvLoc* instances and on maximum number of events that can be located by an *EvLoc* instance to avoid system overload
- Validates and parses input files, populates database tables and builds earth_specs directory tree
- Sends the requester messages associated with *EvLoc* services results, including unauthorized user, invalid input files, success, and out of service messages
- Provides both batch and interactive event location modes
- Applies CMR message tracking system to allow users to refer to previously submitted data
- Generates final results in either IMS1.0 Bulletin or XML format as specified by the user
- Provides a mechanism to monitor *EvLoc* request/process/response status and handle exceptions
- Provides help information such as user manuals, man pages, and information on data exchange formats
- Records usage statistics which can be used to generate reports and high level summaries on *EvLoc* services
- Provides downloadable conversion utilities to generate required input files in XML format

User-provided input

required

- Station data
- Event list with associated arrivals

optional

- *EvLoc* control parameters
- One-dimensional travel time tables
- SSSCs
- SASCs

supported formats

- *LocSAT* input file format for required data (station and event list)
- XML format for required data (station and event list)
- CMR travel-time table file format
- CMR SASC file format
- CMR parameter file format for *EvLoc* control parameters
- Free-format flat files for travel-time tables, SSSCs

Interactivity

- user can change values through the web browser
- safeguards needed to avoid system overload and endless user queues
- may limit number of events/session in interactive use
- no further requirements on client side

user-adjustable parameters

- initial location
- phase names
- defining flags (TAS)
- phase weights (deltim, delslo, delaz)
- *EvLoc* control parameters

Appendix B: New Tables

I: Table descriptions: The necessary changes affect only the database schema. The database table specifics are shown in Table B-1 through Table B-5**.**

Table B-1: elpmonitor - monitor *EvLocProcess* status

field name	field no.	storage type	external format	character positions	field description
elpid	1	i4	i8	1-8	*EvLocProcess* identifier
machine	2	c33	a32	10-41	machine name where *EvLocProcess* resides
pid	3	i4	i8	43-50	process identifier assigned by Operating System
srcid	4	i4	i8	52-59	request source identifier
srcidtype	5	c17	a16	61-76	request source identifier type
status	6	c33	a32	78-109	process status
modtime	7	f8	f17.5	111-127	time of last status change (in epoch)
lddate	8	c18	a17	129-145	load date

Name:	**elpmonitor**	
Keys:	Primary.	elpid
	Alternate.	none
	Foreign.	srcid, srcidtype
Data:	Descriptive.	machine, pid
	Measurement	status, modtime
	Administrative	lddate

Table B-2: elsgrouptype - HLS user group type table

field name	field no.	storage type	external format	character positions	field description
elsgid	1	i4	i8	1-8	HLS user group identifier
elsgtype	2	c17	a16	10-25	HLS user group type
lddate	3	c18	a17	27-43	load date

Name:	**elsgrouptype**	
Keys:	Primary.	elsgid
	Alternate.	none
	Foreign.	none

Data:	Descriptive.	elsgtype
	Administrative	lddate

Table B-3: elssession - HLS session monitoring table

field name	field no.	storage type	external format	character positions	field description
elsgid	1	i4	i8	1-8	HLS user group identifier
userid	2	i4	i8	10-17	user identifier
sessid	3	i4	i8	19-26	session identifier
time	4	f8	f17.5	28-44	start time in epoch
endtime	5	f8	f17.5	46-62	end time in epoch
timeout	6	c2	a1	64-64	yes (y) or no (n) flag to indicate whether the session is timeout or not
status	7	c33	a32	66-97	session status
modtime	8	f8	f17.5	99-115	time of last status change (in epoch)
lddate	9	c18	a17	117-133	load date

Name:	**elssession**	
Keys:	Primary.	elsgid, userid, sessid
	Alternate.	none
	Foreign.	userid
Data:	Descriptive.	time, endtime
	Measurement	timeout, modtime
	Administrative	lddate

Table B-4: elsusage: HLS usage monitoring table

field name	field no.	storage type	external format	character positions	field description
elsgid	1	i4	i8	1-8	HLS user group identifier
userid	2	i4	i8	10-17	user identifier
srcid	3	i4	i8	19-26	request source identifier
srcidtype	4	c17	a16	28-43	request source identifier type
sessid	5	i4	i8	45-52	session identifier
reqtime	6	f4	f17.5	54-70	request time in epoch
refid	7	c17	i8	72-79	request reference identifier
refidtype	8	c17	a16	81-96	request reference identifier type
evloccputime	9	f8	f17.5	98-114	*EvLoc* process CPU time in seconds
totalprotime	10	f8	f17.5	116-132	total process time in second
totalevent	11	i4	i8	134-141	total number of events to be relocated
eventcount	12	i4	i8	143-150	number of successful relocated events
totalarrival	13	i4	i8	152-159	total number of arrivals
par	14	c2	a1	161-161	yes (y) or no (n) flag to indicate whether a parameter file is provided
vmsf	15	c2	a1	163-163	yes (y) or no (n) flag to indicate whether a velocity model specification file is provided
mdf	16	c2	a1	165-165	yes (y) or no (n) flag to indicate whether a magnitude description file is provided
sssc	17	c2	a1	167-167	yes (y) or no (n) flag to indicate whether Source Station Specific Correction files are provided
sasc	18	c2	a1	169-169	yes (y) or no (n) flag to indicate whether Slowness-Azimuth Station Correction files are provided
ttt	19	c2	a1	171-171	yes (y) or no (n) flag to indicate whether Travel Time Table files are provided
inprotocol	20	c17	a16	173-188	input request protocol
outprotocol	21	c17	a16	190-205	output data protocol
dpdir	22	c65	a64	207-270	request input and output data persistent directory
ifsize	23	i4	i8	272-279	input file size in kbytes
lddate	24	c18	a17	281-297	load date

Name:	**elsusage**	
Keys:	Primary.	elsgid, userid, srcid, srcidtype
	Alternate.	none
	Foreign.	elgid, userid, sessid
Data:	Descriptive.	reqtime, refid, refidtype, totalevent, totalarrival, par, vmsf, mdf, sssc, sasc, ttt, inprotocol, outprotocol, dpdir, ifsize

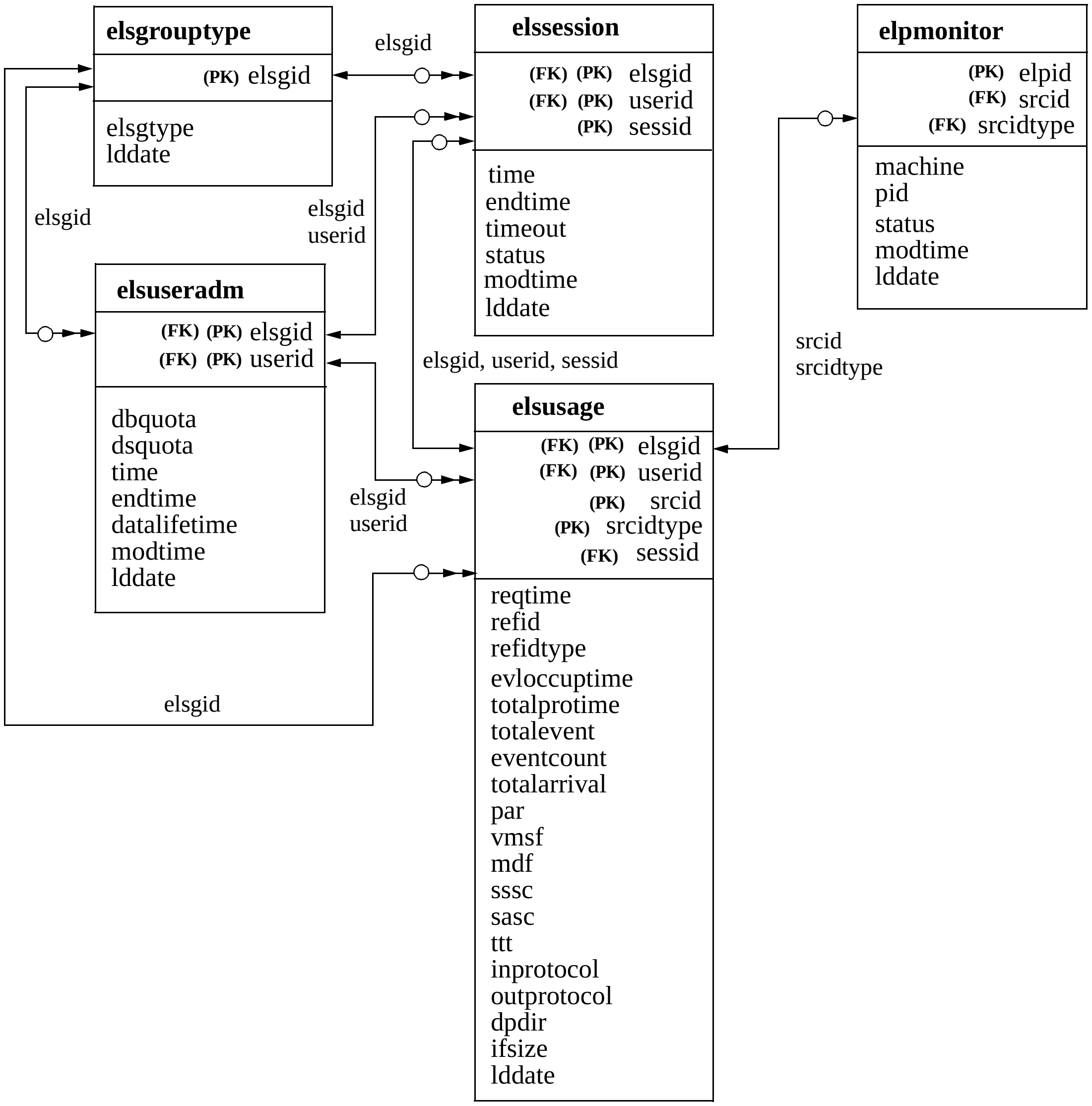
	Measurement	evloccputime, totalprotime, eventcount
	Administrative	lddate

Table B-5: elsuseradm: HLS user administration table

field name	field no.	storage type	external format	character positions	field description
elsgid	1	i4	i8	1-8	HLS group identifier
userid	2	i4	i8	10-17	user identifier
dbquota	3	i4	i8	19-26	database quota in mbytes
dsquota	4	i4	i8	28-35	diskspace quota in mbytes
time	5	f8	f17.5	37-53	start time in epoch
endtime	6	f8	f17.5	55-71	end time in epoch
datalifetime	7	f8	f17.5	73-89	data life time in seconds
modtime	8	c18	a17	91-107	time of last modification
lddate	9	c18	a17	109-125	load date

Name:	**elsuseradm**	
Keys:	Primary.	elsgid, userid
	Alternate.	none
	Foreign.	userid
Data:	Descriptive.	dbquota, dsquota, time, endtime, datalifetime
	Measurement	modtime
	Administrative	lddate

II. Table-relationship:



III Table attributes:

A summary of the database attributes used in the tables is provided here.

1. *datalifetime* Table: **elsuseradm**
Description: Data life time. A new identifier.
Format: float(53)External: f17.5
NA Value: -999.0
Units: seconds
Range: *datalifetime* > 0

2. *dbquota* Table: **elsuseradm**
Description: Database quota. A new identifier.
Format: number(8) External: i8
NA Value: -1
Units: mbytes
Range: *dbquota* > 0

3. *dsquota* Table: **elsuseradm**
Description: Diskspace quota. A new identifier.
Format: number(8) External: i8
NA Value: -1
Units: mbytes
Range: *dsquota* > 0

4. *dpdir* Table: **elsusage**
Description: Request input and output data persistent directory. A new identifier.
Format: varchar2(64) External: a64
NA Value: -
Units: N/A
Range: any string up to 64 characters that conforms to UNIX directory name syntax

5. *elpid* Table: **elpmonitor**
Description: *EvLocProcess* identifier. A new identifier which is the primary key in the **elpmonitor** table.
Format: number(8) External: i8
NA Value: NOT ALLOWED
Units: N/A
Range: *elpid* > 0

6. *elsgid* Table: **elsgrouptype**, **elssession**, **elsusage**, **elsuseradm**
Description: HLS user group identifier. A new identifier, which is the primary key in the **elsgrouptype** table, a member of the composite primary key in the **elssession** table, a member of the composite primary key in the **elsusage** table, a member of the composite primary key in the **elsuseradm** table, a foreign key in the **elssession** table, a foreign key in the **elsusage** table, and a foreign key in the **elsuseradm** table.
Format: number(8) External: i8
NA Value: NOT ALLOWED
Units: N/A
Range: *elsgid* > 0

7. *elsgtype* Table: **elsgrouptype**
Description: HLS user group type. A new identifier
Format: varchar2(16) External: a16

NA Value: -
Units: N/A
Range: any string up to 16 characters

8. *endtime* Table: **elssession**, **elsuseradm**
   Description: Epoch end time. Already defined in schema.

9. *eventcount* Table: **elsusage**
   Description: Number of successful relocated events. A new identifier.
   Format: number(8) External: i8
   NA Value: -1
   Units: N/A
   Range: *eventcount* > 0

10. *evloccputime* Table: **elsusage**
    Description: *EvLoc* process CPU time. A new identifier.
    Format: float(53) External: f17.5
    NA Value: -999.0
    Units: seconds
    Range: *evloccputime* > 0

11. *ifsize* Table: **elsusage**
    Description: Input file size. A new identifier.
    Format: number(8) External: i8
    NA Value: -1
    Units: kbytes
    Range: *ifsize* > 0

12. *inprotocol* Table: **elsusage**
    Description: Input request protocol. A new identifier.
    Format: varchar2(16) External: a16
    NA Value: -
    Units: N/A
    Range: any string up to 16 characters

13. *lddate* Table: **elpmonitor**, **elsgrouptype**, **elssession**, **elsusage**, **elsuseradm**
    Description: Load date. Already defined in schema.

14. *machine* Table: **elpmonitor**
    Description: Machine name where *EvLocProcess* resides. Already defined in schema.

15. *mdf* Table: **elssession**
    Description: Yes (y) or no (n) flag to indicate whether a magnitude description file is provided. A new identifier.
    Format: varchar2(1) External: a1
    NA Value: -
    Units: N/A
    Range: *mdf* E {y, n}

16. *modtime* Table: **elpmonitor**, **elssession**
    Description: Time of last status change. Already defined in schema.

17. *outprotocol* Table: **elsusage**
    Description: Output data protocol. A new identifier.
    Format: varchar2(16) External: a16
    NA Value: -
    Units: N/A
    Range: any string up to 16 characters

18. *par*

Table: **elssession**
Description: Yes (y) or no (n) flag to indicate whether a parameter file is provided. A new identifier.
Format: varchar2(1) External: a1
NA Value: -
Units: N/A
Range: *par* E {y, n}

19. *pid*

Table: **elpmonitor**
Description: Process identifier assigned by Operating System. A new identifier.
Format: number(8) External: i8
NA Value: -1
Units: N/A
Range: *pid* > 0

20. *refid*

Table: **elsusage**
Description: Request reference identifier. Already defined in schema.

21. *refidtype*

Table: **elsusage**
Description: Request reference identifier type. A new identifier.
Format: varchar2(16) External: a16
NA Value: -
Units: N/A
Range: any string up to 16 characters

22. *reqtime*

Table: **elsusage**
Description: Request time. A new identifier.
Format: float(53) External: f17.5
NA Value: -999.0
Units: seconds
Range: *reqtime* > 0

23. *sasc*

Table: **elssession**
Description: Yes (y) or no (n) flag to indicate whether Slowness-Azimuth Station Correction file are provided. A new identifier.
Format: varchar2(1) External: a1
NA Value: -
Units: N/A
Range: *sasc* E {y, n}

24. *sessid*

Table: **elssession, elsusage**
Description: Session identifier. A new identifier, which is a member of the composite primary key in the **elssession** table, and a foreign key in the **elsusage** table.
Format: number(8) External: i8
NA Value: NOT ALLOWED, -1 for **elsusage**
Units: N/A
Range: *sessid* > 0

25. *srcid*

Table: **elpmonitor**, **elsusage**
Description: Request source identifier. A new identifier, which is a member of the composite primary key in the **elsusage** table, and a foreign key in the **elpmonitor** table.
Format: number(8) External: i8
NA Value: -1
Units: N/A
Range: *srcid* > 0

26. *srcidtype*
Table: **elpmonitor**, **elsusage**
Description: Request source identifier type. A new identifier, which is a member of the composite primary key in the **elsusage** table, and a foreign key in the **elpmonitor** table.
Format: varchar2(16) External: a16
NA Value: -
Units: N/A
Range: any string up to 16 characters

27. *sssc*
Table: **elssession**
Description: Yes (y) or no (n) flag to indicate whether Source Station Specific Correction files are provided. A new identifier.
Format: varchar2(1) External: a1
NA Value: -
Units: N/A
Range: *sssc* E {y, n}

28. *status*
Table: **elpmonitor**, **elssession**
Description: Process/session status. Already defined in schema.
Format: varchar2(32) External: a32
NA Value: -
Units: N/A
Range: *status* E {RUNNING, DONE, DONE-ERROR, DONE-SIG}
for **elpmonitor**
status E {ACTIVE, INACTIVE, DONE} for **elssession**

29. *time*
Table: **elssession**, **elsuseradm**
Description: Epoch start time. Already defined in schema.

30. *timeout*
Table: **elssession**
Description: yes (y) or no (n) flag to indicate whether the session is timeout or not. A new identifier.
Format: varchar2(1) External: a1
NA Value: -
Units: N/A
Range: *timeout* E {y, n}

31. *totalarrival*
Table: **elsusage**
Description: Total number of arrivals. A new identifier.
Format: number(8) External: i8
NA Value: -999.0
Units: N/A
Range: *totalarrival* > 0

32. *totalevent*
Table: **elsusage**
Description: Total number of events to be relocated. A new identifier.
Format: number(8) External: i8
NA Value: -999.0
Units: N/A
Range: *totalevent* > 0

33. *totalprotime*
Table: **elsusage**
Description: Total process time. A new identifier.
Format: float(53) External: f17.5
NA Value: -999.0
Units: seconds
Range: *totalprotime* > 0

34. *ttt*

Table: **elssession**
Description: Yes (y) or no (n) flag to indicate whether Travel Time Table files are provided. A new identifier.
Format: varchar2(1) External: a1
NA Value: -
Units: N/A
Range: *ttt* E {y, n}

35. *userid*

Table: **elssession**, **elsusage**, **elsuseradm**
Description: User identifier. Already defined in schema. This identifier is a member of the composite primary key in the **elssession** table, a member of the composite primary key in the **elsusage** table, a member of the composite primary key in the **elsuseradm** table, a foreign key in the **elssession** table, a foreign key in the **elsusage** table, and a foreign key in the **elsuseradm** table.
NA Value: -1

36. *vmsf*

Table: **elssession**
Description: Yes (y) or no (n) flag to indicate whether a velocity model specification file is provided. A new identifier.
Format: varchar2(1) External: a1
NA Value: -
Units: N/A
Range: *wmsf* E {y, n}

Appendix C: Test Results

HLS Test Reports:

The beta tests provide thorough normal and failure tests for seismic, hydroacoustic and infrasonic event location, and magnitude determination. All test results are stored under /home/dev5/SDF/HLS/test folder. All bugs discovered during the test period were fixed in the patch RDSS_1.0.7. We present test results as submitted by beta testers. The test reports indicate that HLS operates as expected, therefore we consider beta tests as a success. Table C-1 provides results of normal process testing and Table C-2 provides results of abnormal process testing.

I. Hypocenter location inversion test:

Date	: March 27, 2002
Author	: **Bernard Hedeline, David Shamos**
File Location	: /home/dev5/SDF/HLS/test/event/DShamos
HLS Comment	: Test results shown in both table C-1 and table C-2 are as expected.

Table C-1: Normal Process Testing

Request Message Type	Results	Comments
LocSAT input file with default *EvLoc* settings.	Correct response with result file provided.	Success
XML format files with default *Evloc* settings.	Correct response with result file provided.	Success
XML format files with customized *EvLoc* parameters.	Correct response with result file provided.	Success
XML format files with customized *EvLoc* parameters, VMSF file, travel time tables, amplitude table for magnitude calculation.	Correct response with result file provided.	Success
XML format files with customized EvLoc parameters, VMSF file, travel time tables, and SSSCs.	Correct response with result file provided.	Success
XML format files with customized *EvLoc* parameters, and SASCs.	Correct response with result file provided.	Success
All parameters and flat files are provided: XML format files, amplitude table, SASCs, Travel Times, VMSF, SSSCs and *EvLoc* parameters.	Correct response with result file provided.	Success

.

Table C-2: Abnormal Process Testing

Request Message Type	Results	Comments
Invalid e-mail message: missing key word "MSG_ID"	Received confirmation message with "request msgid=-" and correct response message.	Normally "request msgid" should have a number.[1]
Invalid e-mail message: missing key word "FTP_FILE"	Received error message: [error] need to specify FTP_FILE	Expected behavior
Invalid e-mail message: incorrect key word "_XML" (should be "SITE_XML"):	Received error message: [error] need to specify either station or site xml file	Expected behavior
Invalid input file name: file name "ml.tar.gz" (should be "xml.tar.gz")	Received error message: [error] unable to retrieve ftp file /gso-public/evloc/ml.tar.gz from host=ftp.saic.com	Expected behavior
Invalid input file name: ASSOC_XML assssoc.xml (should be: assoc.xml	Received error message: [error] failed to open input file assssoc.xml	Expected behavior
Incorrect location path for data input file "xml.tar.gz"	Received error message: [error] unable to retrieve ftp file /gso-public/xml.tar.gz from host=ftp.saic.com	Expected behavior
Data input tar file did not contain TT files and SSSC files which were declared in the e-mail request message: "TTT_FILES iasp91/*" and "SSSC_FILES iasp91/SSSC/*".	Received normal response message with result file. No error was reported.[2]	Expected behavior
All XML files have permission "none" to group and public.	Received normal response message with result file.	Permission "none" to all files for group and public did not affect *EvLoc* processing.
All files sent with request message have permission "none" to group and public.	Received normal response message with result file.	Same as above.
Invalid or unknown format: changed VSMF file to the format not acceptable by CMR.	Received normal response message with result file. No data was processed by *EvLoc*.	Expected behavior

Table C-2: Abnormal Process Testing

Request Message Type	Results	Comments
Invalid or unknown format: changed format of the several TT files.	Received error message: [error] fail to read *EvLoc* par file[info] Finished the process at 2002/03/07 01:46:40.462 (31.432 sec)	Expected behavior
Incorrect data in the TT files iasp91.P, iasp91.Pn	Received normal response message with result file. No data was processed by *EvLoc*.	Expected behavior
Incorrect data in the VSMF file.	Received normal response message with result file. No data was processed by *EvLoc*.	Expected behavior
Incorrect data in the several of the SASC files (sasc.ASAR, sasc.PDY and sasc.ARCES)	Received normal response message with result file. No data was processed by *EvLoc*.	Expected behavior

Tables upper limit: the arrival table had over 7000 tuples.	Received error message:
	[error] try to insert table exceed upper limit -268441140[3]
		According to the developer current limit for all tables is: site=20000
		origin=500
		arrival/assoc/amplitude/stamag = 5000
		netmag=1500

[1] MSG_ID won’t be assigned if the message is not a valid IMS format

[2] default iasp91 travel time and SSSC files will be used in this case

[3] formatting bug, which was fixed in the HLS R1.0

II. Magnitude determination test:

Date	: May 20, 2002
Author	: **Hans Israelsson**
File Location	: /home/dev5/SDF/HLS/test/mag/HIsraelsson
HLS Comment	: HLS provides various types of magnitude determination supported by *EvLoc,* though only four types were tested here.

Testing of *Evloc* Magnitude Calculation

Validation of the *EvLoc* magnitude calculation was performed with spot checks for about 10 REB events (randomly selected), all with mb > 0.

The validation was limited to magnitude types mb_ave, mb1, ms_ave, and ms1.

For a given event and magnitude the following attributes were verified for the stamag table: magtype, sta, ampid, arid, orid, evid, delta, magnitude, uncertainty, magres, phase, magdef, mmodel and for the netmag table:
magtype, orid, evid, nsta, magnitude, uncertainty
Floating numbers were required to within 0.001.

The following four configuration cases were tested:
1. default values
2. non-default values
test of configurable parameter list_of_magtypes
3. non-default mdf file
(a value of 0.123 was added to all mb1 corrections in idc_det.mdf file to verify that *EvLoc* uses the configured file and not the default file)
4. stamag and netmag tables submitted to verify use of parameter use_prev_defs use_prev_defs was set to 0 as outliers are sometimes rejected in the REB manually regardless of whether a station observations fulfills criteria to be included in network magnitude calculation.

In all cases the *EvLoc* parameter mode=1, which means that only magnitudes were calculated and no event location was carried out.

After correcting errors that were detected in the process of testing of *EvLoc* magnitude calculation. The events were re-processed and comparisons with the REB stamag and netmag tables showed that the calculated tables were consistent in accordance with the configuration par files and the REB results.

III. Hydroacoustic event test:

Date	: May 20, 2002
Author	: **David Shamos**
File Location	: /home/dev5/SDF/HLS/test/hydro
HLS Comment	: The test case covers both hydroacoustic and seismic phases.

Hypocenter Location Server *EvLoc* Test Summary Report

During this test event location was computed by using detections at the five hydro stations (H08N, H08S, H04N, H04S, and HA01) and one seismic station (KMBO).

The selected data from the input origin, assoc, arrival and site tables where exported into XML format files. See extract from the par file DBtoXML_ind.par below:

```
table_1=site
tablename_1=site_ims
```

```
table_1_query="select * from site_ims where sta in (select sta from
affiliation_ims_kmbo
where net='IO_HYD')"
table_1_filename="site_ind.xml"

table_1=origin
tablename_1=origin_io_hydro_and_kmbo
table_1_query="select * from origin_io_hydro_and_kmbo where time between
1021612000 and
1021613000"
table_1_filename="origin_ind.xml"

table_1=arrival
tablename_1=arrival_indian_ocean
table_1_query="select * from arrival_indian_ocean where arid in (select arid
from
assoc_io_hydro_and_kmbo where orid in (select orid from
origin_io_hydro_and_kmbo
where time between 1021612000 and 1021613000))"
table_1_filename="arrival_ind.xml"

table_1=assoc
tablename_1=assoc_io_hydro_and_kmbo
table_1_query="select * from assoc_io_hydro_and_kmbo where orid in (select
orid from
origin_io_hydro_and_kmbo where time between 1021612000 and 1021613000)"
table_1_filename="assoc_ind.xml"
```

Note: Database account syngen@sd815

In addition to the listed above XML files the following files were used as an input:

```
1-D hydro TT tables (iasp91/*)
SSSC files (iasp91/SSSC/*)
VMSF file in CMR format (io_acd_test.defs)
EvLoc par file (sssc_ind.par)
```

All input files were archived (e.g. tar.gz) and uploaded to public server ftp.saic.com

The following request message was formed and sent to evloc@cmr.gov:

```
BEGIN IMS1.0
MSG_TYPE PROCESS
MSG_ID 2 SSSCs, VMSF, TT, custom parfile
ORIGIN_XML origin_ind.xml
ARRIVAL_XML arrival_ind.xml
ASSOC_XML assoc_ind.xml
SITE_XML site_ind.xml
TTT_FILES iasp91/*
SSSC_FILES iasp91/SSSC/*
VMSF_FILE io_acd_test.defs
PAR_FILE sssc_ind.par
```

```
FTP_FILE ftp.saic.com anonymous /gso-public/evloc ind_ocean_acd_2.tar.gz
TIME_STAMP
EvLoc XML
STOP
```

Confirmation message was received:

```
[info] Your request msgid=2 was received at 2002/05/20 17:58:55
[info] The tracking number is msgid=749
```

Response message followed the confirmation:

```
BEGIN IMS1.0
MSG_TYPE DATA
MSG_ID 750 RDSS
REF_ID 749 msgid
DATA_TYPE ftp_log
FTP_FILE ftp.pidc.org anonymous /pub/kung els_msgid_749.zip
% or simply click on ftp://ftp.pidc.org/pub/kung/els_msgid_749.zip
% file els_msgid_749.zip will be removed after 7 days
TIME_STAMP 2002/05/20 18:01:36
STOP
```

The results were downloaded from ftp.cmr.gov. See directory
/home/mist/Testing/automatic/EvLoc_interface/Working/results/msgid_749

The test output shows that by using the listed above input data interactive version of *EvLoc* was able to locate an event with detections at the five hydro stations and one seismic station.

See extract from the log fileEvLoc_749.log below:

```
  Final location estimate (+/- S.D.):
      Latitude:         0.131 deg. S +/-    18.958 km.
     Longitude:        60.053 deg. E +/-    15.879 km.
         Depth:         0.000  km.
 Relative O.T.:      -309.528 sec.   +/-     2.128 sec.
 Absolute O.T.: 1021612799.981 sec.   +/-     2.128 sec.
              : May 17 2002 05:19:59.98
```

The SQL query below shows the original origin lat, lon, depth, and time before the *EvLoc* processing:

```
SQL> select LAT, LON, DEPTH, TIME
  2  from origin_io_hydro_and_kmbo
  3  where time between 1021612000 and 1021613000;

      LAT       LON       DEPTH            TIME
--------- --------- ----------- ---------------
```

```
    .0000   60.0000       .0000  1021612800.000
```

IV. Infrasonic/hydroacoustic event test:

Date	: June 5, 2002
Author	: **Anna Gault Galjan**
File Location	: /home/dev5/SDF/HLS/test/infra
HLS Comment	: If only three defined phases are provided, we recommend to set the *EvLoc* parameter “fix_depth=1” to prevent from getting *EvLoc* error message “Insufficient defining data before location is even attempted.”

A number of hydroacoustic and infrasonic events were tested against the new HLS. These events test the robustness of the HLS and *EvLoc*, since they involve relatively new types of phases (’H’ and ’I’), and because these events tend to be very small, with a limited number of phases. The acoustic events were tested by submitting site, origin, arrival, and assoc information, and a par file to evloc@cmr.gov. The par file contained only verbosity and depth parameters.

The first tests using hydroacoustic and infrasonic events from the archive database failed. These events had been taken from the REB, where they had been successfully located in the past. A number of errors were reported in the log files, including the messages:

"Locator: Condition number of solution is too great to continue!"
"Locator: Insufficient defining data before location is even attempted!"

Two problems were found and corrected. First, the software had not been configured to recognize the ’H’ and ’I’ phases. Second, the depth had not been fixed to the surface, which is necessary for hydroacoustic and infrasonic events. The HLS default *EvLoc* par file was fixed to recognize ’H’ and ’I’ phases, and the user-defined *EvLoc* par file was corrected to fix the depth to the surface. Once these fixes were applied, the re-submitted acoustic events successfully located, even the very small 3-station hydroacoustic event with only time-defining phases.

The following are two examples of successful re-tests using a hydroacoustic event and an infrasonic event:

A 5 station event with all time- and azimuth-defining phases, corresponding to REB origin 21243107, was successfully located at latitude 29.282 deg. N +/- 52.475 km; longitude 135.118 deg. W +/- 63.754 km, and depth of 0.000 km. There were no errors reported.

A very small hydroacoustic event with 3 time-defining phases, corresponding to REB origin 1187824, was successfully located at latitude 37.659 deg. N +/-   15.356 km, longitude 123.304 deg. W +/- 26.166 km, and depth 0.000 km. There were no errors reported.

Users will notice complaints from *EvLoc* that no SSSC information was found for any of the acoustic stations; these warnings do not interfere with the operation of *EvLoc* and should be ignored.

Appendix D: Sample Parameter Files

Table D-1 lists parameter files required for HLS. Those parameter files and associated manual pages are included in the releases RDSS_1.0.3 and RDSS_1.0.7.

Table D-1: HLS Parameter files

Parameter (Par) file	**Application**	**Par File Status**	**Comment**
`MessageGet.par`	*MessageGet*	revised from OPS' par file	might replace OPS' MessageGet.par in the future
`HLS_setup.par`	*HLS_setup.pl*	new	
`EvLocProcess.par`	*EvLocProcess*	new	
`EvLoc_def.par`	*EvLoc*	new (derived from OPS' `EvLoc.par` file)	used only in the HLS as a default *EvLoc* par file
`user_EvLoc.par`	*EvLoc*	new (derived from OPS' `EvLoc.par` file)	used only in the HLS as a default user-defined *EvLoc* par file
`HLS_cleanup.par`	*HLS_cleanup.pl*	new	
`DBtoXML.par`	*DBtoXML*	new	used on client side
`XMLtoDB.par`	*XMLtoDB*	new	used on client side

The new parameters revised from OPS' parameter files are shown in italics below. We also recommend to add five global variables to the $IMSPAR file. Both `EvLoc_def.par` and `user_EvLoc.par` are rearranged from OPS' `EvLoc.par` file in order to suit the needs of the HLS. There are no new parameters in either file.

`$IMSPAR`: add following variables

```
# HLS dir
ELS_ROOT=( proper directory )
ELS_BIN=$(ELS_ROOT)/bin
ELS_CONFIG_DIR=$(ELS_ROOT)/config
ELS_EXPORT_DIR=$(ELS_ROOT)/export
ELS_DEFAULT_EARTH_SPECS=$(ELS_CONFIG_DIR)/earth_specs
```

`EvLoc_def.par:`

```
#
```

```
# EvLoc default parameter file
#
par=$(IMSPAR)

ELSDB=account/password@dbserver

in_db_account=$(ELSDB)
out_db_account=$(ELSDB)
db_vendor=$(DATABASE_VENDOR)

vmodel_spec_file=$(VMSF)
mag_descrip_file=$(MDF)
tl_spec_file=$(TLSF)
sasc_dir_prefix=$(SASC_DIR_PREFIX)

network=CUR_IDC
origin_query="where 1=1"

#
# default input tables
#

site_table=site
affiliation_table=affiliation
origin_table=origin
assoc_table=assoc
arrival_table=arrival
det_amplitude_table=amplitude
ev_amplitude_table=amplitude
parrival_table=parrival
netmag_table=netmag
stamag_table=stamag


#
# default output tables
#

write_to_input_db_tables=0
new_origin_table=new_origin
new_assoc_table=new_assoc
new_origerr_table=new_origerr
new_netmag_table=new_netmag
new_stamag_table=new_stamag


#
# write output ar_info table. The ar_info table contains extended
#    associated-based information which includes individual contributions
#    to the travel-time (elevation correction, ellipticity correction, bulk
#    static station correction, source-dependent correction) model and
#    measurement errors, data importances and slowness vector residual
write_ar_info_table=1
ar_info_table=ar_info
```

```
#
# some flags
#

write_af_tables=0
triple_location=0
use_ev_cntrl_table=0
write_ev_cntrl_table=0
create_syn_data_only=0

#
# ----- Primary event location control parameters -----
#

#
# --- The following lines are ADSN-specific locator control variables ---
#

ignore_big_res=0
big_res_mult=3.0
use_tscor=0


# Ellipticity correction level:
#    0 = No correction
#    1 = ADSN ellipticity correction
#    2 = IASPEI 1991 ellipticity correction
ellip_cor_level=2


#
# User defined EvLoc par file
#

# Control mode
#    0: Event location only (default)
#    1: Magnitude only
#    2: Both location and magnitude
mode=0

# Phase list for IASPEI (1991) T-T tables
list_of_phases="Lg,P,P3KPbc,P3KPbc_B,P3KPdf,P3KPdf_B,PKKPab,PKKPbc,PKKPdf,PKKS
ab,PKKSbc,PKKSdf,PKPab,PKPbc,PKPdf,PKP2ab,PKP2bc,PKP2df,PKSab,PKSbc,PKSdf,PKiK
P,PP,PP_B,PPP,PPP_B,PPS,PPS_B,PS,PcP,PcS,Pg,Pn,Rg,S,SKKPab,SKKPbc,SKKPdf,SKKSa
c,SKKSdf,SKPab,SKPbc,SKPdf,SKSdf,SKSac,SKS2ac,SKS2df,SKiKP,SP,SS,SS_B,SSS,SSS_
B,ScS,ScP,Sn,pP,pPKPab,pPKPbc,pPKPdf,pS,pSKSac,pSKSdf,sP,sPKPab,sPKPbc,sPK-
Pdf,sPKiKP,sS,sSKSac,sSKSdf,H,I,T,O"

#
# ----- Primary event location control parameters -----
#

# Level of verbosity for printed locator output. Scaled )
# from 0 (no printed output to 4 (all output printed).
verbose=1

# Fix (constrain) event origin time
fix_ot=0

# Fix (constrain) event latitude & longitude
fix_latlon=0

# Fix (constrain) event hypocentral depth
fix_depth=0

# Fixed (constrained) event hypocentral depth (km)
depth_init=-999.

# Confidence level for computing error ellipses
# (only 0.80, 0.90, 0.95 and 0.99 currently permitted)
conf_level=0.90

# A priori number of degrees of freedom to define between
# "confidence" (=0) and "coverage" (= very large number) ellipse
num_dof=99999

# Percent damping to be applied to diagonal elements of system matrix.
# If negative, locator will determined its own damping, if necessary.
damp=-1.00

# Maximum number of iterations allowed
max_iter=20

# SSSC level:
#    0 = No SSSC desired
#    1 = Regional SSSC only, if available
#    2 = Local SSSC, if available
sssc_level=0

# whether apply pre-defined distance modeling errors as stored
# within the travel-time tables
dist_var_wgt=1

# Use only stations with an SSSC, SRST or test-site corrections
# in determing event location.
only_sta_w_corr=0

# Use only stations specified immediately below in calculating
# an event location?
sub_sta_list_only=0
sub_sta_list=""



#
# ----- Primary event magnitude control parameters -----
#
```

```
mag_num_boots=20
mag_ignore_big_res=0
mag_big_res_mult=3.0
mag_sub_sta_list_only=0
mag_sub_sta_list="AS12,BC03,BM05,CM16,FX01,IL01,IM03,TT01"
mag_use_tsco=0
#mag_ts_region=AA
mag_only_sta_w_corr=0

# Level of verbose magnitude output to be printed:
#    0: None
#    1: Network mag info only
#    2: Network and station mag info
mag_verbose=0

# Specify desired list of magnitude types (magtypes)
list_of_magtypes="mb_ave"
list_of_mb_magtypes="mb_ave"
magtype_to_origin_mb="mb"
magtype_to_origin_ms="ms"
magtype_to_origin_ml="ml"

# Use magdef settings for all pre-existing stamag records.
use_prev_magdefs=0
```

user_EvLoc.par:

```
#
# EvLoc user-specified parameters list
#

# --- Primary control parameters ---


mode=0# Control mode [2]
#   0: Event location only
#   1: Magnitude only
#   2: Both location and magnitude (default)

# If mode = 0 or 2, you must specify a list of acceptable phase types
verbose=1

# Phase list for IASPEI (1991) and AK135 (1995) T-T tables
list_of_phases="Lg,P,P3KPbc,P3KPbc_B,P3KPdf,P3KPdf_B,PKKPab,PKKPbc,PKKPdf,PKKS
ab,PKKSbc,PKKSdf,PKPab,PKPbc,PKPdf,PKP2ab,PKP2bc,PKP2df,PKSab,PKSbc,PKSdf,PKiK
P,PP,PP_B,PPP,PPP_B,PPS,PPS_B,PS,PcP,PcS,Pg,Pn,Rg,S,SKKPab,SKKPbc,SKKPdf,SKKSa
c,SKKSdf,SKPab,SKPbc,SKPdf,SKSdf,SKSac,SKS2ac,SKS2df,SKiKP,SP,SS,SS_B,SSS,SSS_
B,ScS,ScP,Sn,pP,pPKPab,pPKPbc,pPKPdf,pS,pSKSac,pSKSdf,sP,sPKPab,sPKPbc,sPK-
Pdf,sPKiKP,sS,sSKSac,sSKSdf,H,I,T"

# --- Primary event location control parameters ---

fix_depth=0# Fix (constrain) event hypocentral depth [1=TRUE]
```

```
#depth_init=0.0# Fixed (constrained) event hypocentral depth (km) [0.0]

conf_level=0.90# Confidence level for computing error ellipses (Only
# 0.80, 0.90, 0.95 and 0.99 currently permitted)  [0.90]

sssc_level=0# 0 = No SSSC desired; 1 = Regional SSSC only, if
# available; and 2 = Local SSSC, if available.  [0]

# Use only stations with an SSSC, SRST or test-site corrections
# in determing event location.
only_sta_w_corr=0

# --- The following lines are ADSN-specific locator control variables ---
dist_var_wgt=1# Apply model errors [TRUE]

sub_sta_list_only=0     # Use only stations specified immediately below in
                        # calculating an event location ?       [FALSE]
sub_sta_list=""

# --- Primary event magnitude control parameters ---

# --- Specify desired list of magnitude types (magtypes) desired:
list_of_magtypes="mb,mlppn"
list_of_mb_magtypes="mb_ave"# List of mb magtypes only for handling mb-only
# controls (i.e., min/max distance & mmodel)
use_prev_magdefs=0# Use magdef settings for all pre-existing
# stamag records.[FALSE]
mag_verbose=0
```

MessageGet.par:

```
#CCASEID: @(#) MessageGet.par CMR /main/9  04/17/2001
# %W% %G%
#
par=$(IMSPAR)
par=$(MESSAGES)

database=$(EXPERTDB)
vendor=$(DATABASE_VENDOR)

#   Incoming messages go through the system on the basis of status
#   RECEIVED -> QUEUED -> RUNNING -> FAILED or STANDBY
status=RECEIVED
status-new=RECEIVED
status-queued=QUEUED
status-running=RUNNING
status-child_failed=FAILED

flush-queued=1
max-queued=9
cautious=0
sleep-time=5
```

```
loop=1
ipc=False

#   Incoming messages separated into different queues based on MSGTYPE
#   and SUBTYPE.  In this configuration there is 1 valid queue, plus
#   a catch-all queue (MessageAlert).
message-type-0=request
message-type-1=data
message-type-2=FTP-LOG
message-type-3=request-l
message-type-4=request-w
message-type-5=request-r
message-type-6=subscription
message-type-7=process

#   For each queue, gives maximum number of active children.
max-running-0=1
max-running-1=5
max-running-2=1
max-running-3=1
max-running-4=2
max-running-5=1
max-running-6=1
max-running-7=1

#   For each queue, gives the name of the executable.
request-path=$(RELBIN)/AutoDRM
data-path=$(RELBIN)/ParseData
FTP-LOG-path=$(RELBIN)/MessageFTP
request-l-path=$(RELBIN)/AutoDRM
request-w-path=$(RELBIN)/AutoDRM
request-r-path=$(RELBIN)/AutoDRM
process-path=$(RELBIN)/EvLocProcess
subscription-path=$(RELBIN)/ParseSubs
other-path=$(RELBIN)/MessageAlert

#    For each queue, gives the single command line argument.
request-par=$(PARDIR)/AutoDRM/AutoDRM.par
data-par=$(PARDIR)/ParseData/ParseData.par
FTP-LOG-par=$(PARDIR)/MessageFTP/MessageFTP.par
request-l-par=$(PARDIR)/AutoDRM/AutoDRM.par
request-w-par=$(PARDIR)/AutoDRM/AutoDRM.par
request-r-par=$(PARDIR)/AutoDRM/AutoDRM_r.par

subscription-par=$(MESSAGES-DIR)/ParseSubs/ParseSubs.par
process-par=$(ELS_CONFIG_DIR)/app_config/els/EvLocProcess.par
other-par=$(MESSAGES-DIR)/MessageAlert/MessageAlert.par

data-ignore-subtype
FTP-LOG-ignore-subtype
subscription-ignore-subtype
other-ignore-subtype

log-directory=$(MSGLOGDIR)
```

```
log-name=MessageGet.log
log-files=20
log-lines=100
log-misc
```

Appendix E: HLS User Guide

The HLS user guide is located at CMR RDSS web site. The current web address is https://www.cmr.gov/rdss/hls/UserGuide.html. The file CCB0207_AppendixE.html is the snapshot of the HLS user guide on July 12, 2002.

Appendix F: HLS Examples Document

The HLS examples document is located at CMR RDSS web site. The current web address is https://www.cmr.gov/rdss/hls/Examples.html. The file CCB0207_AppendixF.html is the snapshot of the HLS examples document on July 12, 2002.